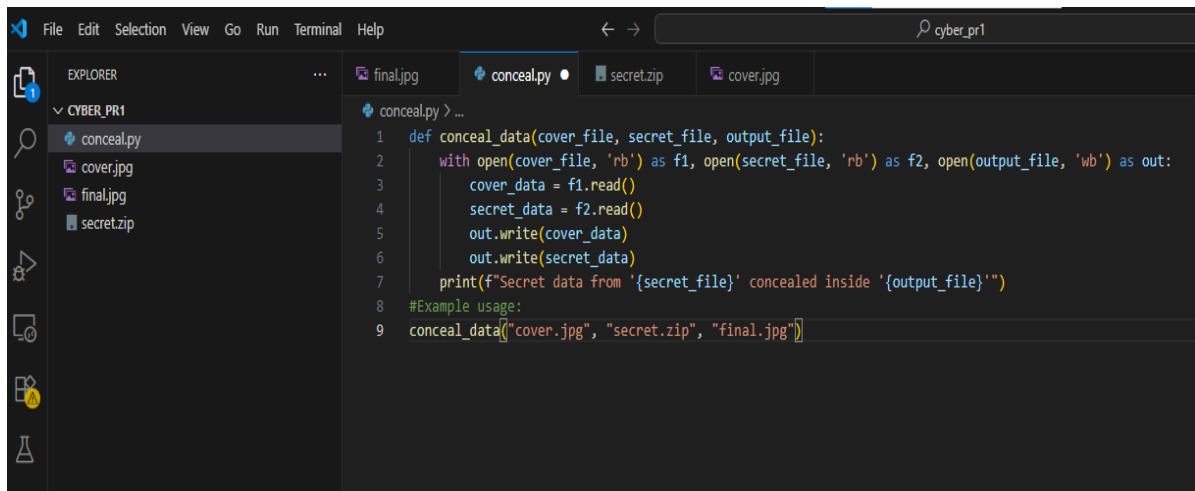
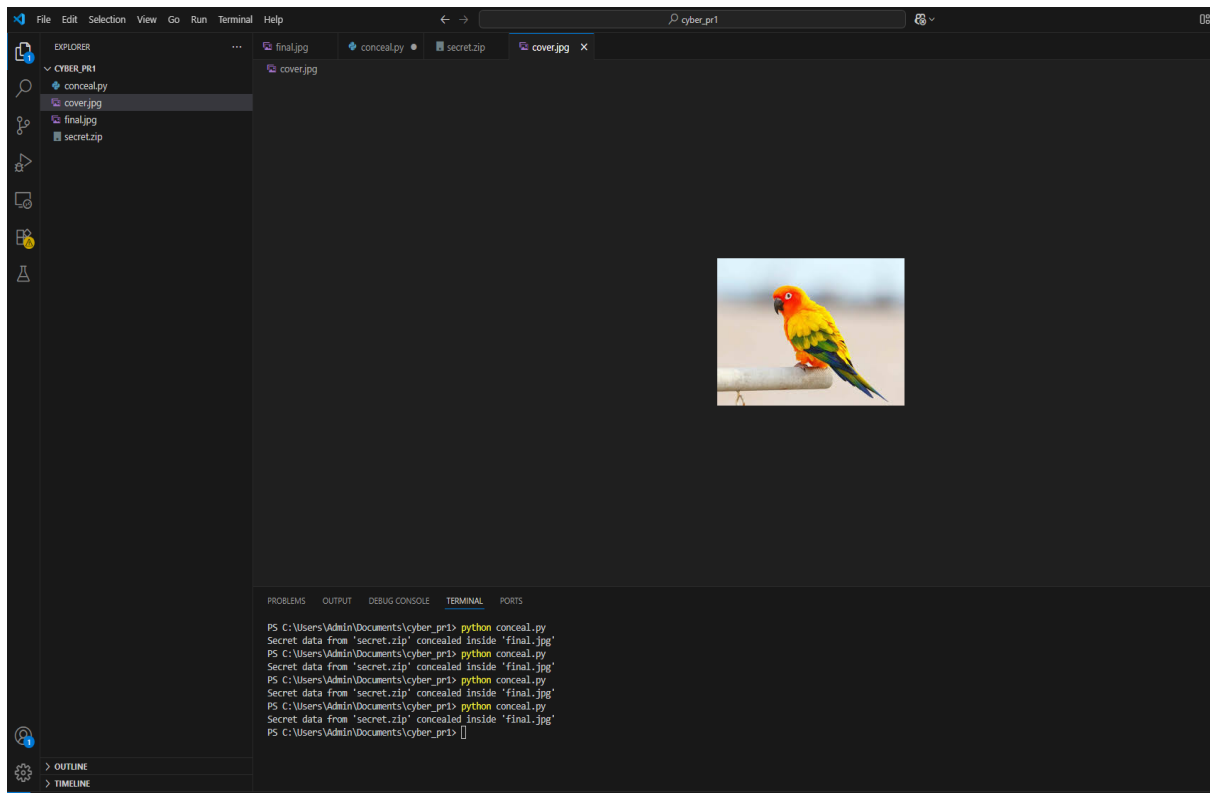


CYBER_PR1



The screenshot shows the Visual Studio Code editor interface. The Explorer pane on the left displays a project named 'CYBER_PR1' containing four files: 'conceal.py', 'cover.jpg', 'final.jpg', and 'secret.zip'. The 'conceal.py' file is selected and its code is visible in the main editor area. The code defines a function 'conceal_data' that takes three arguments: 'cover_file', 'secret_file', and 'output_file'. It uses 'with open' to read data from the cover and secret files and write it to the output file. An example usage is provided at the bottom of the script.

```
1 def conceal_data(cover_file, secret_file, output_file):
2     with open(cover_file, 'rb') as f1, open(secret_file, 'rb') as f2, open(output_file, 'wb') as out:
3         cover_data = f1.read()
4         secret_data = f2.read()
5         out.write(cover_data)
6         out.write(secret_data)
7     print(f"Secret data from '{secret_file}' concealed inside '{output_file}'")
8 #Example usage:
9 conceal_data("cover.jpg", "secret.zip", "final.jpg")
```



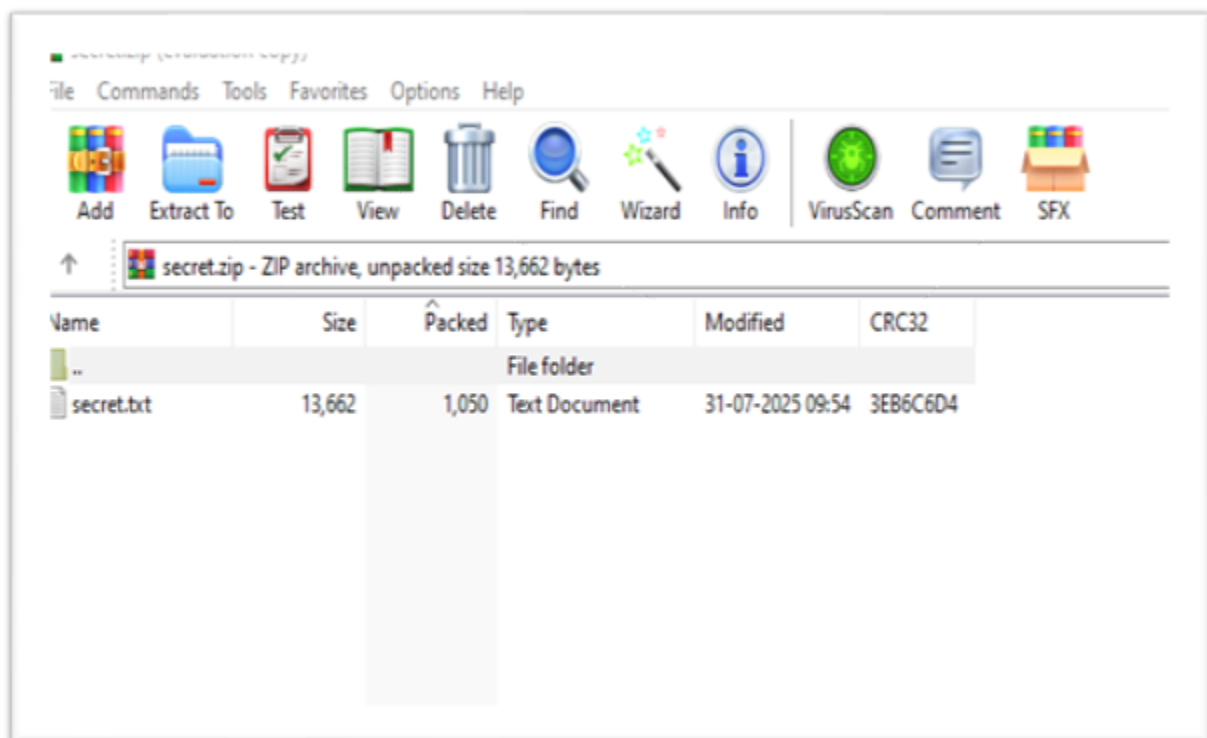
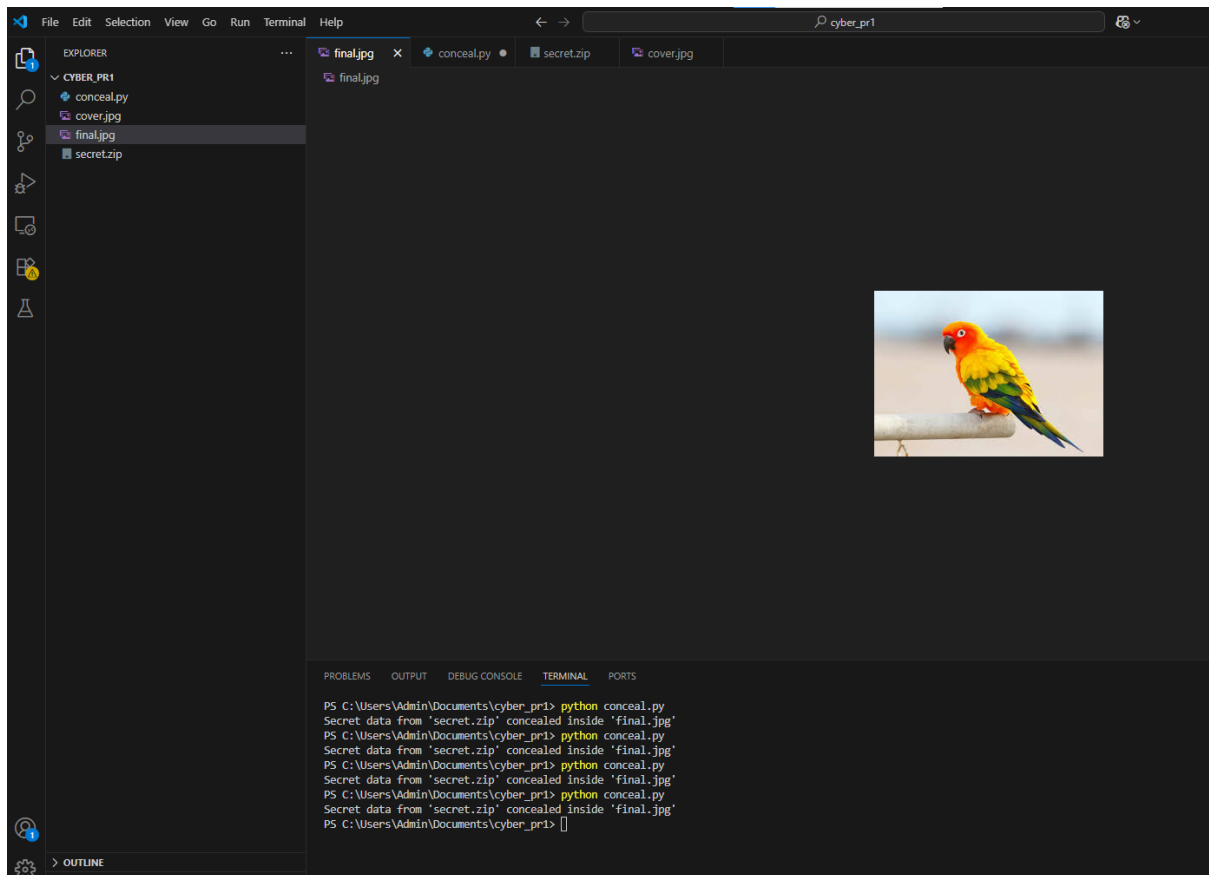


Image Steganography

[How it works](#)

[How to Defend it](#)

Hide images inside other images.

This is a client-side Javascript tool to steganographically hide images inside the lower "bits" of other images.

Select either "Hide image" or "Unhide image". Play with the **example** images (all 200x200 px) to get a feel for it.

Hide image

Unhide image

Cover image:

Choose File

Example:



Secret image:

Choose File

Example:



Hidden bits: 7

Download Full-size Image



This is a project by [James Staley](#).

You can learn more about [Steganography](#) on [Wikipedia](#).